

# Free Lunch or No Free Lunch: That is not Just a Question?

Xin-She Yang

Mathematics and Scientific Computing

National Physical Laboratory, Teddington, TW11 0LW, UK

## Abstract

The increasing popularity of metaheuristic algorithms has attracted a great deal of attention in algorithm analysis and performance evaluations. No-free-lunch theorems are of both theoretical and practical importance, while many important studies on convergence analysis of various metaheuristic algorithms have proven to be fruitful. This paper discusses the recent results on no-free-lunch theorems and algorithm convergence, as well as their important implications for algorithm development in practice. Free lunches may exist for certain types of problem. In addition, we will highlight some open problems for further research.

**Citation Details:** X. S. Yang, Free lunch or no free lunch: that is not just a question? *Int. J. Artificial Intelligence Tools*, Vol. 21, No. 3, 1240010 (2012). DOI: 10.1142/S0218213012400106

# 1 Introduction

Metaheuristic algorithms form an important part of contemporary global optimization algorithms [1, 2, 5, 45, 26, 31]. Good examples are simulated annealing and particle swarm optimization [24, 25]. They work remarkably efficiently and have many advantages over traditional, deterministic methods and algorithms, and thus they have been applied in almost all area of science, engineering and industry [9, 11, 10, 48].

Despite such a huge success in applications, mathematical analysis of algorithms remains limited and many open problems are still un-resolved. There are three challenging areas for algorithm analysis: complexity, convergence and no-free-lunch theory.

Complexity analysis of traditional algorithms such as quick sort and matrix inverse are well-established, as these algorithms are deterministic. In contrast, complexity analysis of metaheuristics remains a challenging task, as the stochastic nature of these algorithms. However, good results do exist, concerning randomization search techniques [1].

Convergence analysis is another challenging area. One of the main difficulties concerning the convergence analysis of metaheuristic algorithms is the lack of a generic framework, though substantial studies have been carried out using dynamic systems and Markov processes. However, convergence analysis still remains one of the active research areas with many encouraging results [6, 39, 30, 17].

Along the mathematical analysis of optimization algorithms, another equally challenging, and yet fruitful area is the theory on algorithm performance and comparison, leading to a wide range of no-free-lunch (NFL) theorems [42, 22]. While in well-posed cases of optimization functional space in finite domains, NFL theorems do hold, however, free lunches are possible [1, 43, 40].

In this paper, we will briefly review and summarize the recent studies of no-free-lunch theory and also free lunch scenario. This enable us to view the NLF and free lunch in a unified framework, or at least, in a convenient way. We will also briefly highlights some of the convergence studies. Based on these studies, we will summarize and propose a series of recommendations for further research.

## 2 No-Free-Lunch Theorems

The seminal paper by Wolpert and Mcready in 1997 essentially proposed a framework for performance comparison of optimization algorithms, using a combination of Bayesian statistics and Markov random field theories.

Along many relevant assumptions in proving the NFL theorems, two fundamental assumptions are: finite states of the search space (and thus the objective values), and the non-revisiting time-ordered sets.

The first assumption is a good approximation to many problems, especially in finite-digit approximations. However, there is mathematical difference in countable finite, and countable infinite. Therefore, the results for finite states/domains may not directly applicable to infinite domains. Furthermore, as continuous problem are un-

countable, NFL results for finite domains will usually not hold for continuous domains [1].

The second assumption on non-revisiting iterative sequence is an over-simplification, as almost all metaheuristic algorithms are revisiting in practice, some points visited before will possibly be re-visited again in the future. The only possible exception is the Tabu algorithm with a very long Tabu list [13]. Therefore, results for non-revisiting time-ordered iterations may not be true for the cases of revisiting cases, because the revisiting iterations break an important assumption of ‘closed under permutation’ (c.u.p) required for proving the NFL theorems [28].

Furthermore, optimization problems do not necessarily concern the whole set of all possible functions/problems, and it is often sufficient to consider a subset of problems. It is worth pointing out active studies have carried out in constructing algorithms that can work best on specific subsets of optimization problems, in fact, NFL theorems do not hold in this case [8].

Before we go further to discuss more about any possible free lunches, let us sketch Wolpert and Macready’s original proof. Assuming that the search space is finite (though quite large), thus the space of possible objective values is also finite. This means that objective function is simply a mapping  $f : \mathcal{X} \mapsto \mathcal{Y}$ , with  $\mathcal{F} = \mathcal{Y}^{\mathcal{X}}$  as the space of all possible problems under permutation.

As an algorithm tends to produce a series of points or solutions in the search space, it is further assumed that these points are distinct. That is, for  $k$  iterations,  $k$  distinct visited points forms a time-ordered set

$$\Omega_k = \left\{ \left( \Omega_k^x(1), \Omega_k^y(1) \right), \dots, \left( \Omega_k^x(k), \Omega_k^y(k) \right) \right\}. \quad (1)$$

There are many ways to define a performance measure, though a good measure still remains debatable [35]. Such a measure can depend on the number of iteration  $k$ , the algorithm  $a$  and the actual cost function  $f$ , which can be denoted by  $P(\Omega_k^y | f, k, a)$ . Here we follow the notation style in seminal paper by Wolpert and Mcready (1997). For any pair of algorithms  $a$  and  $b$ , the NFL theorem states

$$\sum_f P(\Omega_k^y | f, k, a) = \sum_f P(\Omega_k^y | f, k, b). \quad (2)$$

In other words, any algorithm is as good (bad) as a random search, when the performance is averaged over all possible functions.

Wolpert and Macready’s original proof was carried out by induction. Using a similar methodology and similar assumptions, other forms of NFL theorems can also be proved. These theorems are vigorous and thus have important theoretical values. However, their practical implications are a different issue. In fact, it may not be so important in practice anyway, we will discuss this in a later section.

## 3 Free Lunch or No Free Lunch

### 3.1 Continuous Free Lunches

The validity of NFL theorems largely depends on the validity of their fundamental assumptions. However, whether these assumptions are valid in practice is another question. Often, these assumptions are too stringent, and thus free lunches are possible.

One of the assumptions is the non-revisiting nature of the  $k$  distinct points which form a time-ordered set. For revisiting points as they do occur in practice in real-world optimization algorithms, the ‘closed under permutation’ does not hold, which renders NFL theorems invalid [33, 28]. This means free lunches do exist in practical applications.

Another basic assumption is the finiteness of the domains. For continuous domains, Auger and Teytaud in 2010 have proven that the NFL theorem does not hold, and therefore they concluded that “continuous free lunches exist”. Indeed, some algorithms are better than others. For example, for a 2D sphere function, they demonstrated that an efficient algorithm only needs 4 iterations/steps to reach the global minimum.

### 3.2 Coevolutionary and Multiobjective Free Lunches

The basic NFL theorems concern a single agent, marching iteratively in the search space in distinct steps. However, Wolpert and Mcready proved in 2005 that NFL theorems do not hold under coevolution. For example, a set of players (or agents) in self-play problems can work together so as to produce a champion. This can be visualized as an evolutionary process of training a chess champion. In this case, free lunch does exist [43]. It is worth pointing out that for a single player, it tries to pursue the best next move, while for two players, the fitness function depends on the moves of both players. Therefore, the basic assumptions for NFL theorems are no longer valid.

For multiobjective optimization problems, things have become even more complicated. An important step in theoretical analysis is that some multiobjective optimizers are better than others as pointed out by Corne and Knowles [7]. One of the major reasons is that the archiver and generator in the multiobjective algorithms can lead to multiobjective free lunches.

Whether NFL holds or not, it has nothing to say about the complexity of the problems. In fact, no free lunch theorem has not been proved to be true for problems with NP-hard complexity [41].

## 4 Practical Implications of NFL Theorems

No-free-lunch theorems may be of theoretical importance, and they can also have important implications for algorithm development in practice, though not everyone agrees the real importance of these implications.

There are three kinds of opinions concerning the implications. The first group may simply ignore these theorems, as they argue that the assumptions are too stringent, and the performance measures based on average overall functions are irrelevant in practice. Therefore, no practical importance can be inferred, and research just carries on.

The second kind is that NFL theorems can be true, and they can accept that the fact there is no universally efficient algorithm. But in practice some algorithms do performance better than others for a specific problem or a subset of problems. Research effort should focus on finding the right algorithms for the right type of problem. Problem-specific knowledge is always helpful to find the right algorithm(s).

The third kind of opinion is that NFL theorems are not true for other types of problems such as continuous problems and NP-hard problems. Theoretical work concerns more elaborate studies on extending NFL theorems to other cases or on finding free lunches [1]. On the other hand, algorithm development continues to design better algorithms which can work for a wider range of problems, not necessarily all types of problems. As we have seen from the above analysis, free lunches do exist, and better algorithms can be designed for a specific subset of problems [46, 49].

Thus, free lunch or no free lunch is not just a simple question, it has important and yet practical importance. There is certain truth in all the above arguments, and their impacts on optimization community are somehow mixed. Obviously, in reality, the algorithms with problem-specific knowledge typically work better than random search, and we also realized that there is no universally generic tool that works best for all the problems. Therefore, we have to seek balance between speciality and generality, between algorithm simplicity and problem complexity, and between problem-specific knowledge and capability of handling black-box optimization problems.

## 5 Convergence Analysis

For convergence analysis, there is no mathematical framework in general to provide insights into the working mechanism, the complexity, stability and convergence of any given algorithm [19, 38]. Despite the increasing popularity of metaheuristics, mathematical analysis remains fragmental, and many open problems concerning convergence analysis need urgent attention.

### 5.1 PSO

Particle swarm optimization (PSO) was developed by Kennedy and Eberhart in 1995 [24], based on the swarm behaviour such as fish and bird schooling in nature. Since then, PSO has generated much wider interests, and forms an exciting, ever-expanding research subject, called swarm intelligence. PSO has been applied to almost every area in optimization, computational intelligence, and design/scheduling applications.

The movement of a swarming particle consists of two major components: a stochastic component and a deterministic component. Each particle is attracted

toward the position of the current global best  $\mathbf{g}^*$  and its own best location  $\mathbf{x}_i^*$  in history, while at the same time it has a tendency to move randomly.

Let  $\mathbf{x}_i$  and  $\mathbf{v}_i$  be the position vector and velocity for particle  $i$ , respectively. The new velocity and location updating formulas are determined by

$$\mathbf{v}_i^{t+1} = \mathbf{v}_i^t + \alpha \epsilon_1 [\mathbf{g}^* - \mathbf{x}_i^t] + \beta \epsilon_2 [\mathbf{x}_i^* - \mathbf{x}_i^t]. \quad (3)$$

$$\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + \mathbf{v}_i^{t+1}, \quad (4)$$

where  $\epsilon_1$  and  $\epsilon_2$  are two random vectors, and each entry taking the values between 0 and 1. The parameters  $\alpha$  and  $\beta$  are the learning parameters or acceleration constants, which can typically be taken as, say,  $\alpha \approx \beta \approx 2$ .

There are at least two dozen PSO variants which extend the standard PSO algorithm, and the most noticeable improvement is probably to use inertia function  $\theta(t)$  so that  $\mathbf{v}_i^t$  is replaced by  $\theta(t)\mathbf{v}_i^t$  where  $\theta \in [0, 1]$ . This is equivalent to introducing a virtual mass to stabilize the motion of the particles, and thus the algorithm is expected to converge more quickly.

The first convergence analysis of PSO was carried out by Clerc and Kennedy in 2002 [6] using the theory of dynamical systems. Mathematically, if we ignore the random factors, we can view the system formed by (3) and (4) as a dynamical system. If we focus on a single particle  $i$  and imagine that there is only one particle in this system, then the global best  $\mathbf{g}^*$  is the same as its current best  $\mathbf{x}_i^*$ . In this case, we have

$$\mathbf{v}_i^{t+1} = \mathbf{v}_i^t + \gamma(\mathbf{g}^* - \mathbf{x}_i^t), \quad \gamma = \alpha + \beta, \quad (5)$$

and

$$\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + \mathbf{v}_i^{t+1}. \quad (6)$$

Considering the 1D dynamical system for particle swarm optimization, we can replace  $\mathbf{g}^*$  by a parameter constant  $p$  so that we can see if or not the particle of interest will converge towards  $p$ . By setting  $u_t = p - x(t+1)$  and using the notations for dynamical systems, we have a simple dynamical system

$$v_{t+1} = v_t + \gamma u_t, \quad u_{t+1} = -v_t + (1 - \gamma)u_t, \quad (7)$$

or

$$Y_{t+1} = AY_t, \quad A = \begin{pmatrix} 1 & \gamma \\ -1 & 1 - \gamma \end{pmatrix}, \quad Y_t = \begin{pmatrix} v_t \\ u_t \end{pmatrix}. \quad (8)$$

The general solution of this dynamical system can be written as  $Y_t = Y_0 \exp[At]$ . The system behaviour can be characterized by the eigenvalues  $\lambda$  of  $A$

$$\lambda_{1,2} = 1 - \frac{\gamma}{2} \pm \frac{\sqrt{\gamma^2 - 4\gamma}}{2}. \quad (9)$$

It can be seen clearly that  $\gamma = 4$  leads to a bifurcation.

Following a straightforward analysis of this dynamical system, we can have three cases. For  $0 < \gamma < 4$ , cyclic and/or quasi-cyclic trajectories exist. In this case, when randomness is gradually reduced, some convergence can be observed. For  $\gamma > 4$ ,

non-cyclic behaviour can be expected and the distance from  $Y_t$  to the center  $(0, 0)$  is monotonically increasing with  $t$ . In a special case  $\gamma = 4$ , some convergence behaviour can be observed. For detailed analysis, please refer to Clerc and Kennedy [6]. Since  $p$  is linked with the global best, as the iterations continue, it can be expected that all particles will aggregate towards the the global best.

## 5.2 Firefly Algorithm

Firefly Algorithm (FA) was developed by Yang [45, 47], which was based on the flashing patterns and behaviour of fireflies. In essence, each firefly will be attracted to brighter ones, while at the same time, it explores and searches for prey randomly. In addition, the brightness of a firefly is determined by the landscape of the objective function.

The movement of a firefly  $i$  is attracted to another more attractive (brighter) firefly  $j$  is determined by

$$\mathbf{x}_i = \mathbf{x}_i + \beta_0 e^{-\gamma r_{ij}^2} (\mathbf{x}_j - \mathbf{x}_i) + \alpha \boldsymbol{\epsilon}_i, \quad (10)$$

where the second term is due to the attraction. The third term is randomization with  $\alpha$  being the randomization parameter, and  $\boldsymbol{\epsilon}_i$  is a vector of random numbers drawn from a Gaussian distribution or uniform distribution. Here is  $\beta_0 \in [0, 1]$  is the attractiveness at  $r = 0$ , and  $r_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|$  is the Cartesian distance. For other problems such as scheduling, any measure that can effectively characterize the quantities of interest in the optimization problem can be used as the ‘distance’  $r$ . For most implementations, we can take  $\beta_0 = 1$ ,  $\alpha = O(1)$  and  $\gamma = O(1)$ . It is worth pointing out that (10) is essentially a random walk biased towards the brighter fireflies. If  $\beta_0 = 0$ , it becomes a simple random walk. Furthermore, the randomization term can easily be extended to other distributions such as Lévy flights.

We now can carry out the convergence analysis for the firefly algorithm in a framework similar to Clerc and Kennedy’s dynamical analysis. For simplicity, we start from the equation for firefly motion without the randomness term

$$\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + \beta_0 e^{-\gamma r_{ij}^2} (\mathbf{x}_j^t - \mathbf{x}_i^t). \quad (11)$$

If we focus on a single agent, we can replace  $\mathbf{x}_j^t$  by the global best  $g$  found so far, and we have

$$\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + \beta_0 e^{-\gamma r_i^2} (g - \mathbf{x}_i^t), \quad (12)$$

where the distance  $r_i$  can be given by the  $\ell_2$ -norm  $r_i^2 = \|g - \mathbf{x}_i^t\|_2^2$ . In an even simpler 1-D case, we can set  $y_t = g - \mathbf{x}_i^t$ , and we have

$$y_{t+1} = y_t - \beta_0 e^{-\gamma y_t^2} y_t. \quad (13)$$

We can see that  $\gamma$  is a scaling parameter which only affects the scales/size of the firefly movement. In fact, we can let  $u_t = \sqrt{\gamma} y_t$  and we have

$$u_{t+1} = u_t [1 - \beta_0 e^{-u_t^2}]. \quad (14)$$

These equations can be analyzed easily using the same methodology for studying the well-known logistic map

$$u_t = \lambda u_t(1 - u_t). \quad (15)$$

Straightforward analysis can show that convergence can be achieved for  $\beta_0 < 2$ . There is a transition from periodic to chaos at  $\beta_0 \approx 4$ . This may be surprising, as the aim of designing a metaheuristic algorithm is to try to find the optimal solution efficiently and accurately. However, chaotic behaviour is not necessarily a nuisance, in fact, we can use it to the advantage of the firefly algorithm. Simple chaotic characteristics from (15) can often be used as an efficient mixing technique for generating diverse solutions. Statistically, the logistic mapping (15) for  $\lambda = 4$  for the initial states in (0,1) corresponds a beta distribution

$$B(u, p, q) = \frac{\Gamma(p+q)}{\Gamma(p)\Gamma(q)} u^{p-1} (1-u)^{q-1}, \quad (16)$$

when  $p = q = 1/2$ . Here  $\Gamma(z)$  is the Gamma function

$$\Gamma(z) = \int_0^\infty t^{z-1} e^{-t} dt. \quad (17)$$

In the case when  $z = n$  is an integer, we have  $\Gamma(n) = (n-1)!$ . In addition,  $\Gamma(1/2) = \sqrt{\pi}$ . From the algorithm implementation point of view, we can use higher attractiveness  $\beta_0$  during the early stage of iterations so that the fireflies can explore, even chaotically, the search space more effectively. As the search continues and convergence approaches, we can reduce the attractiveness  $\beta_0$  gradually, which may increase the overall efficiency of the algorithm. Obviously, more studies are highly needed to confirm this.

### 5.3 Markov Chains

Most theoretical studies use Markov chains/process as a framework for convergence analysis. A Markov chain is said to be regular if some positive power  $k$  of the transition matrix  $\mathbf{P}$  has only positive elements. A chain is ergodic or irreducible if it is aperiodic and positive recurrent, which means that it is possible to reach every state from any state.

For a time-homogeneous chain as  $k \rightarrow \infty$ , we have the stationary probability distribution  $\pi$ , satisfying

$$\pi = \pi \mathbf{P}, \quad (18)$$

thus the first eigenvalue is always 1. This will lead to the asymptotic convergence to the global optimality  $\theta_*$ :

$$\lim_{k \rightarrow \infty} \theta_k \rightarrow \theta_*, \quad (19)$$

with probability one [12, 30, 17].

Now if look at the PSO closely using the framework of Markov chain Monte Carlo, each particle in PSO essentially forms a Markov chain, though this Markov chain is biased towards to the current best, as the transition probability often leads



to the acceptance of the move towards the current global best. Other population-based algorithms can also be viewed in this framework. In essence, all metaheuristic algorithms with piecewise, interacting paths can be analyzed in the general framework of Markov chain Monte Carlo. The main challenge is to realize this and to use the appropriate Markov chain theory to study metaheuristic algorithms. More fruitful studies will surely emerge in the future.

## 5.4 Convergence of SA

Simulated annealing and generalize hill-climber algorithms were among the first algorithms with important results on convergence analysis [29, 14, 23, 44, 37]. The main idea is to consider simulated annealing as sequence of homogeneous Markov chains or a long single inhomogeneous Markov chain [20]. Under weak ergodic conditions, the temperature  $T_k$  can be reduced sufficiently slow to zero by

$$T_k \geq \frac{A}{\log(k)}, \quad \lim_{k \rightarrow \infty} T_k \rightarrow 0, \quad (20)$$

where  $A$  is a constant

## 5.5 Convergence of GA

One of the well-studied and most popular algorithms is the class of genetic algorithms [21, 26]. Earlier seminal papers proved the convergence of genetic algorithms [18, 32]. Important studies on convergence analysis of GA have been carried out by Aytug et al. [3, 4], Greenhalgh and Marshal [15], Gutjahr [16, 17] and others.

For example, a well-known result is that the number of iterations  $t(\zeta)$  in GA with a convergence probability of  $\zeta$  can be estimated by the upper limit

$$t(\zeta) \leq \left\lceil \frac{\ln(1 - \zeta)}{\ln \left\{ 1 - \min[(1 - \mu)^{Ln}, \mu^{Ln}] \right\}} \right\rceil,$$

where parameter  $\mu$  is the mutation rate in genetic algorithms.  $L$  and  $n$  are the string length and population size, respectively [3]. These results are further elaborated by others [15].

## 5.6 Multiobjective Metaheuristics

Convergence analysis for single-objective optimization tends to be challenging, this complexity is further complicated by the Pareto optimality of multiobjective optimization. Despite these challenges, asymptotic convergence of metaheuristic, multiobjective optimization has been proved by Villalobos-Arias et al. (2005) using a framework of Markov chains [40]. They proved that the transition matrix  $\mathbf{P}$  of a metaheuristic algorithm can have a stationary distribution  $\pi$  such that

$$|P_{ij}^k - \pi_j| \leq (1 - \zeta)^{k-1}, \quad \forall i, j, \quad (k = 1, 2, \dots),$$

where  $\zeta$  is a function of mutation probability  $\mu$ , string length  $L$  and population size  $n$ . For example, when the population can be divided into two sets with mutation rates  $\mu_1, \mu_2$  and population sizes  $n, n_1$ , respectively, this  $\zeta$  function becomes

$$\zeta = 2^{nL} \mu_1^{n_1 L} \mu_2^{(n-n_1)L}. \quad (21)$$

They demonstrated that an algorithm satisfying this condition may not converge for multiobjective optimization problems, however, an algorithm with *elitism* indeed converges under the above conditions.

## 5.7 Other results

Limited results on convergence analysis exist, concerning finite domains, ant colony optimization [16, 34], cross-entropy optimization, best-so-far convergence [27, 17], nested partition method, Tabu search, and largely combinatorial optimization. However, more challenging tasks for infinite states/domains and continuous problems. Many, many open problems need satisfactory answers.

On the other hand, it is worth pointing out that an algorithm can converge, but it may not be efficient, as its convergence rate could be typically low. One of the main tasks in research is to find efficient algorithms for a given type of problem.

## 6 Open Problems

Active research on NFL theorems and algorithm convergence analysis has led to many important results. Despite this, many crucial problems remain unanswered. These open questions span a diverse range of areas. Here we highlight a few but relevant open problems.

*Framework:* Convergence analysis has been fruitful, however, it is still highly needed to develop a unified framework for algorithmic analysis and convergence.

*Exploration and exploitation:* Two important components of metaheuristics are exploration and exploitation or diversification and intensification. What is the optimal balance between these two components?

*Performance measure:* To compare two algorithms, we have to define a measure for gauging their performance [36]. At present, there is no agreed performance measure, but what are the best performance measures? Statistically?

*Free lunches:* No-free-lunch theorems have not been proved for continuous domains for multiobjective optimization. For single-objective optimization, free lunches are possible, is this true for multiobjective optimization? In addition, no free lunch theorem has not been proved to be true for problems with NP-hard complexity (Whitley and Watson 2005). If free lunches exist, what are their implications in practice and how to find the best algorithm(s)?

*Knowledge:* Problem-specific knowledge always helps to find an appropriate solution? How to quantify such knowledge?

*Intelligent algorithms:* A major aim for algorithm development is to design better, intelligent algorithms for solving tough NP-hard optimization problems. What do

mean by ‘intelligent’? What are the practical ways to design truly intelligent, self-evolving algorithms?

## References

- [1] Auger, A. and Teytaud, O., Continuous lunches are free plus the design of optimal optimization algorithms, *Algorithmica*, **57**, 121-146 (2010).
- [2] Auger, A. and B. Doerr, B., *Theory of Randomized Search Heuristics: Foundations and Recent Developments*, World Scientific, (2010).
- [3] Aytug, H., Bhattacharrya, S., and G. J. Koehler, G. J., A Markov chain analysis of genetic algorithms with power of 2 cardinality alphabets, *Euro. J. Operational Research*, **96**, 195-201 (1996).
- [4] Aytug, H. and Koehler, G. J., New stopping criterion for genetic algorithms, *Euro. J. Operational research*, **126**, 662-674 (2000).
- [5] Blum, C. and Roli, A. (2003) Metaheuristics in combinatorial optimisation: Overview and conceptual comparison, *ACM Comput. Surv.*, Vol. 35, 268-308.
- [6] Clerc, M. and J. Kennedy, J., The particle swarm - explosion, stability, and convergence in a multidimensional complex space, *IEEE Trans. Evolutionary Computation*, **6**, 58-73 (2002).
- [7] Corne D. and Knowles, J., Some multiobjective optimizers are better than others, *Evolutionary Computation*, CEC’03, **4**, 2506-2512 (2003).
- [8] Christensen, S. and Oppacher, F., (2001). Wath can we learn from No Free Lunch? in: *Proc. Genetic and Evolutionary Computation Conference* (GECCO-01), pp. 1219-1226 (2001).
- [9] Cui, Z. H., and Zeng, J. C. (2005) A modified particle swarm optimisation predicted by velocity, GECCO 2005, pp. 277–278.
- [10] Cui, Z. H., and Cai, X. J. (2009) Integral particle swarm optimisation with dispersed accelerator information, *Fundam. Inform.*, Vol. 95, 427–447.
- [11] Floudas, C. A. and Pardalos, P. M., *Encyclopedia of Optimization*, 2nd Edition, Springer (2009).
- [12] Gamerman, D., *Markov Chain Monte Carlo*, Chapman & Hall/CRC, (1997).
- [13] Glover, F. and Laguna, M. (1997) *Tabu Search*, Kluwer Academic Publishers, Boston.
- [14] Granville, V., Kravanek, M. and Rasson, J. P., Simulated annealing: a proof of convergence, *IEEE Trans. Pattern Aanlysis and Machine Intelligence*, **16**, 652-656 (1994).

- [15] Greenhalgh, D. and S. Marshal,S., Convergence criteria for genetic algorithms, *SIAM J. Computing*, **30**, 269-282 (2000).
- [16] Gutjahr, W. J., ACO algorithms with guaranteed convergence to the optimal solution, *Information Processing Letters*, **82**, 145-153 (2002).
- [17] Gutjahr, W. J., Convergence Analysis of Metaheuristics *Annals of Information Systems*, **10**, 159-187 (2010).
- [18] Hartl, R. F., A global convergence proof for a class of genetic algorithms, Technical Report, University of Vienna, (1990).
- [19] He, J. and Yu, X., Conditions for the convergence of evolutionary algorithms, *J. Systems Architecture*, **47**, 601-612 (2001).
- [20] Henderson, D., Jacobson, S. H., and Johnson, W., The theory and practice of simulated annealing, in: *Handbook of Metaheuristics* (Eds. F. Glover and G. A. Kochenberger), Kluwer Academic, pp. 287-319 (2003).
- [21] Holland, J., *Adaptation in Natural and Artificial systems*, University of Michigan Press, Ann Arbor, (1975).
- [22] Igel, C. and Toussaint, M., (2003). On classes of functions for which no free lunch results hold, *Inform. Process. Lett.*, **86**, 317-321 (2003).
- [23] Jacobson, S. H., Sullivan, K. A., Johnson, A. W., Discrete manufacturing process design optimization using computer simulation and generalized hill climbing algoirthms, *Engineering Optimization*, **31**, 147-260 (1998).
- [24] Kennedy, J. and Eberhart, R. (1995) ‘Particle swarm optimisation’, in: *Proc. of the IEEE Int. Conf. on Neural Networks*, Piscataway, NJ, pp. 1942-1948.
- [25] Kirkpatrick, S., Gellat, C. D., and Vecchi, M. P. (1983) ‘Optimisation by simulated annealing’, *Science*, **220**, 671-680.
- [26] Neumann, F. and Witt, C., *Bioinspired Computation in Combinatorial Optimization: Algorithms and Their Computational Complexity*, Springer, (2010).
- [27] Margolin, L., On the convergence of the cross-entropy method, *Annals of Operations Research*, **134**, 201-214 (2005).
- [28] Marshall, J. A. and Hinton, T. G., Beyond no free lunch: realistic algorithms for arbitrary problem classes, WCCI 2010 IEEE World Congress on Computational Intelligence, July 1823, Barcelona, Spain, pp. 1319-1324 (2010).
- [29] Mitra, D., Romeo, F., and Sangiovanni-Vincentelli, A. L., Convergence and finite time behavior of simulated annealing, *Advances in Applied Probability*, **18**, 747-771 (1986).

- [30] Ólafsson, S., Metaheuristics, in: *Handbook on Simulation* (Eds. Nelson and Henderson), Handbooks in Operation Research and Management Science VII, Elsevier, pp. 633-654 (2006).
- [31] Parpinelli, R. S., and Lopes, H. S., New inspirations in swarm intelligence: a survey, *Int. J. Bio-Inspired Computation*, **3**, 1-16 (2011).
- [32] Rudolph, G., Convergence analysis of canonical genetic algorithms, *IEEE Trans. Neural Networks*, **5**, 96-100 (1994).
- [33] Schumacher, C., Vose, M., and Whitley D., The no free lunch and problem description length, in: *Genetic and Evolutionary Computation Conference, GECCO-2001*, pp. 565-570 (2001).
- [34] Sebastiani, G. and Torrisi, G. L., An extended ant colony algorithm and its convergence analysis, *Methodology and Computing in Applied Probability*, **7**, 249-263 (2005).
- [35] Shilane, D., Martikainen, J., Dudoit, S., Ovaska, S. J. (2008) 'A general framework for statistical performance comparison of evolutionary computation algorithms', *Information Sciences*, Vol. 178, 2870-2879.
- [36] Spall, J. C., Hill, S. D., and Stark, D. R., Theoretical framework for comparing several stochastic optimization algorithms, in: *Probabilistic and Randomized Methods for Design Under Uncertainty*, Springer, London, pp. 99-117 (2006).
- [37] Steinhofel, L., Albrecht A. and Wong, C. K., Convergence analysis of simulated annealing-based algorithms solving flow shop scheduling problems, *Lecture Notes in Computer Science*, **1767**, 277-290 (2000).
- [38] Thikomirov, A. S., On the convergence rate of the Markov homogeneous monotone optimization method, *Computational Mathematics and Mathematical Physics*, **47**, 817-828 (2007).
- [39] Trelea, I. C., The particle swarm optimization algorithm: convergence analysis and parameter selection, *Information Processing Letters*, **85**, 317-325 (2003).
- [40] Villalobos-Arias, M., Coello Coello, C. A., and Hernández-Lerma, O., Asymptotic convergence of metaheuristics for multiobjective optimization problems, *Soft Computing*, **10**, 1001-1005 (2005).
- [41] Whitley, D. and Watson, J. P., Complexity theory and the no free lunch theorem, in: *Search Methodologies*, pp. 317-339 (2005).
- [42] Wolpert, D. H. and Macready, W. G. (1997), No free lunch theorems for optimisation, *IEEE Transaction on Evolutionary Computation*, Vol. 1, 67-82.
- [43] Wolpert, D. H. and Macready, W. G., Coevolutionary free lunches, *IEEE Trans. Evolutionary Computation*, **9**, 721-735 (2005).

- [44] Yang, R. L., Convergence of the simulated annealing algorithm for continuous global optimization, *J. Optimization Theory and Applications*, **104**, 691-716 (2000).
- [45] Yang, X. S. (2008), *Nature-Inspired Metaheuristic Algorithms*, Luniver Press.
- [46] Yang, X. S. (2009), Firefly algorithms for multimodal optimisation, *Proc. 5th Symposium on Stochastic Algorithms, Foundations and Applications, SAGA 2009*, Eds. O. Watanabe and T. Zeugmann, Lecture Notes in Computer Science, Vol. 5792, 169-178.
- [47] Yang, X. S. (2010a), Firefly algorithm, stochastic test functions and design optimisation, *Int. J. Bio-Inspired Computation*, **2**, 78–84.
- [48] Yang, X. S. (2010b), *Engineering Optimization: An Introduction with Meta-heuristic Applications*, John Wiley and Sons, USA (2010).
- [49] Yang X. S. and Deb, S. (2010) Engineering optimisation by cuckoo search, *Int. J. Math. Modelling & Num. Optimisation*, Vol. 1, 330-343.